พงษ์ศักดิ์ หิรัณยรัชต์

ก้าวแรกในการสร้าง มาโครเอ็กเซล ตอนที่ 2

แนะนำวิธีที่ใช้สร้างมาโคร ให้มีประสิทธิภาพ และง่ายต่อการใช้งาน

ในการเขียนมาโครให้มีประสิทธิภาพนั้น ขึ้นอยู่กับองค์ประกอบ 2 ประการคือ ตัวคำสั่งมาโครเอง และเวอร์กชีตที่ใช้งานร่วมกับมาโคร องค์ประกอบทั้งสองนี้เรียกว่า โมเดล (Model) หนังสือ Microsoft Excel User's Guide เล่มที่ 2 ได้แนะนำหลักการที่สำคัญในการสร้างโมเดลให้มีประสิทธิภาพดังนี้คือ

1. แนวทางในการออกแบบมาโคร 3 ประการ
2. การเขียนคำสั่งในแนวดิ่งหรือ Modular Design
3. แนวทางในการสร้างสูตรสำหรับการคำนวณ
4. แนวทางในการสร้างโมเดลให้ง่ายต่อการใช้
5. วางแผนการออกแบบชุดคำสั่งมาโคร

1. แนวทางในการออกแบบมาโคร 3 ประการ

1. **มีคำตอบที่ต้องการ** โมเดลที่สร้างขึ้นจะต้องสามารถหาคำตอบที่ถูกต้องได้ค่อนข้างแน่นอน ในบางครั้งมาโครที่สร้างขึ้นอาจไม่สามารถหาคำตอบได้ไม่ว่าจะด้วยเหตุผลใดก็ตาม (เช่น ป้อนข้อมูลผิด เป็นต้น) ไม่ต้องกังวลใจในเรื่องนี้ หน้าที่ของผู้สร้างโมเดลก็คือสร้างเงื่อนไขขึ้นมาเพื่อดักเอาไว้ เมื่อพบความไม่ชอบมาพากลก็สั่งให้เครื่องแสดงข้อความผิดพลาด (Error Message) ทำให้เราไม่ต้องเสียเวลารอคำตอบโดยเปล่าประโยชน์

2. **แก้ไขปรับปรุงได้ง่าย** โมเดลที่สร้างมีโอกาสที่ต้องได้รับการแก้ไขปรับปรุงต่อไปในอนาคต พยายามออกแบบโมเดลให้มีความสะดวกและ

ง่ายแก่การแก้ไขปรับปรุงที่อาจเกิดขึ้น เพราะไม่แน่ว่าเราเองหรือผู้อื่นเป็นผู้ปรับปรุง

3. ความเร็ว เมื่อเปรียบเทียบความเร็วกับความถูกต้องแล้ว ความเร็วกลายเป็นรองไปโดยปริยาย แต่ก็ไม่เชิงว่าความเร็วในการทำงานไม่มีความสำคัญเลยเสียทีเดียว ขอเปรียบเทียบเพื่อให้ผู้อ่านเห็นภาพดังนี้ มีมาโครอยู่ 2 ชุด ให้ผลลัพธ์เหมือนกันทุกประการ แต่แตกต่างกันตรงที่มาโครชุดแรกมีความยาวมากกว่ามาโครชุดที่สอง หรือมีจำนวนบรรทัดมากกว่านั่นเอง เวลาในการประมวลผล (Execute) ย่อมแตกต่างกัน มาโครชุดแรกจะใช้เวลามากกว่าชุดที่สอง ดังนั้นในการสร้างโมเดลให้มีประสิทธิภาพความเร็วจึงเป็นองค์ประกอบหนึ่งที่ไม่สามารถละเลยได้

2. การเขียนคำสั่งในแนวดิ่ง หรือ Modular Design

ควรออกแบบให้เวอร์กชีตหรือมาโครชีตเลื้อยลงในแนวดิ่งมากกว่าขยายไปด้านข้าง ในการนี้ควรใช้ 1 หรือ 2 คอลัมน์เท่านั้น ประโยชน์ที่ได้จากการนี้ก็คือการคำนวณรวดเร็วขึ้นและง่ายต่อการตรวจสอบ

พยายามแบ่งการคำนวณเป็นขั้นตอนย่อย ๆ โดยจัดให้ขั้นตอนเหล่านั้นอยู่ในแนว (ดิ่ง) เดียวกัน แต่ละขั้นตอนควรทำหน้าที่เพียงอย่างเดียวเท่านั้น เช่นในการคิดภาษีเงินได้บุคคลธรรมดา การคำนวณอาจแบ่งเป็นขั้นตอนได้ดังนี้

1) คำนวณหาเงินได้พึงประเมิน
2) การคำนวณหาค่าใช้จ่าย
3) การคำนวณหาค่าลดหย่อน
4) การคำนวณภาษี

การจัดแบ่งขั้นตอนที่ถือว่ามีประสิทธิภาพก็คือ ในแต่ละขั้นตอนย่อยมีความยาวเท่า ๆ กันหรือมีจำนวนบรรทัดใกล้เคียงกัน และไม่ควรมีความยาวเกิน 1 ถึง 2 หน้าจอภาพ ส่วนที่อยู่ใกล้กันควรแยกให้เห็นเด่นชัดด้วยการฟอร์แมตเซล หรือการใช้กำหนดสีให้แตกต่างกัน หรือการตีกรอบล้อมรอบแต่ละส่วน ลำดับขั้นตอนการคำนวณควรเรียงจากบนลงมาล่าง ซึ่งจะมีผลให้การคำนวณซ้ำ (Recalculation) รวดเร็วขึ้น

	A	B	C
1	ชื่อ ShortCut	ชื่อมาโคร	หมายเหตุ
2	Ctrl + C	**ตอกเบี้ยทบต้น**	
3			
4	InterestRate	=INPUT("ป้อนอัตราดอกเบี้ย",1)	รับค่าอัตราดอกเบี้ย
5	Payment	=INPUT("ป้อนเงินต้น",1)	รับค่าเงินต้น
6	Time	=INPUT("ระยะเวลากู้ยืม",1)	รับค่าระยะเวลากู้ยืม
7	First	=(1+InterestRate/100)	คำนวณเทอมแรก
8	Second	=(First)^Time	คำนวณเทอมที่สอง
9	Answer	=Payment*Second	คำนวณหาคำตอบ
10		=RETURN()	
11			

รูปที่ 1 แสดงการกระจายการคำนวณที่ซับซ้อนเป็นส่วนย่อย ๆ แทนที่จะเก็บสูตรไว้ในเซลเพียงเซลเดียว

3. แนวทางในการสร้างสูตรสำหรับการคำนวณ

การเขียนสูตรการคำนวณให้ง่ายต่อการอ่านหรือตีความ ทำให้ลดความผิดพลาดที่เกิดจากคำนวณ ทำให้โมเดลง่ายต่อการบำรุงรักษา การคำนวณซ้ำรวดเร็วขึ้นเพราะไม่ต้องคำนวณในส่วนที่ไม่จำเป็นต้องทำ มีหลักการอยู่ 3 ประการ

1. กระจายการคำนวณที่ยุ่งยากไปไว้หลาย ๆ เซล แทนที่จะเก็บไว้ในเซลเดียว ทำให้โมเดลง่ายต่อการเข้าใจและง่ายต่อการตรวจสอบข้อผิดพลาด (Error) ที่อาจเกิดขึ้น เช่น สูตรการหาดอกเบี้ยทบต้น (I.C.) คือ $P(1+i)^n$ เมื่อ P คือเงินต้น, i คืออัตราดอกเบี้ย, n คือระยะเวลาที่กู้ยืม ดังแสดงในรูปที่ 1

2. ควรเขียนข้อความกำกับสูตร เพื่อความสะดวกในการตรวจสอบที่มาของสูตร เช่นที่เซล B4 เป็นสูตรในการคำนวณหารายได้สุทธิ ดังนั้นที่เซล A3 ควรระบุว่า รายได้สุทธิ ดังแสดงในรูปที่ 2

3. สูตรคำนวณควรใช้ชื่อตัวแปรแทนเซลแอดเดรส การใช้ชื่อตัวแปรในสูตรคำนวณจะสามารถสื่อความเข้าใจได้ดีกว่าการใช้เซลแอดเดรส ทำให้การพัฒนาที่อาจเกิดขึ้นในอนาคตไม่เสียเวลาในการค้นหาที่มาของสูตร เช่นต้องการหาราคาขาย สูตรการคำนวณควรเป็น =Cost+Profit แทนที่จะเป็น = A50+A55 เป็นต้น

4. แนวทางในการสร้างโมเดลให้ง่ายต่อการใช้

การออกแบบให้โมเดลมีความง่ายต่อการใช้งานโดยเฉพาะกับผู้ใช้มือใหม่แล้ว เป็นเรื่องที่ผู้สร้างโมเดลควรให้ความสนใจ และต่อไปนี้เป็นข้อแนะนำในการสร้างมาโครให้ง่ายต่อการใช้

1. มีหัวข้อเรื่อง/คำแนะนำในการใช้งาน/อธิบายส่วนประกอบของโมเดลไว้ที่ส่วนบนของเวอร์กชีต คำแนะนำในการใช้งานถ้าสามารถสร้างเป็นรูปกราฟิกได้ก็จะเป็นการดี

2. ใช้สีในการแยกความแตกต่าง ผู้เขียนเข้าใจว่าผู้ใช้คอมพิวเตอร์ส่วนใหญ่ในปัจจุบันคงใช้จอภาพแบบสีกันเป็นส่วนใหญ่ ดังนั้นข้อแนะนำข้อนี้ไม่น่าเป็นอุปสรรค ควรใช้สีในการแบ่งแยกให้ผู้ใช้เห็นว่าส่วนไหนสามารถแก้ไขได้ แล้วระบุข้อแตกต่างนี้ไว้ที่บริเวณคำแนะนำในการใช้งานในหัวข้อดังที่กล่าวไว้แล้วในข้อ 1 เช่นบริเวณที่เป็นสีแดงไม่สามารถแก้ไขได้ เป็นต้น

3. ทำการโปรเทคต์เซลที่ไม่อนุญาตให้มีการแก้ไข เพื่อป้องกันไม่ให้ผู้ใช้แก้ไขส่วนที่เป็นหัวใจของการทำงาน

4. ไม่ควรแสดง Grid และ Row หรือ Column Heading จะทำให้เวอร์กชีตหรือมาโครชีตมองดูไม่เกะกะ

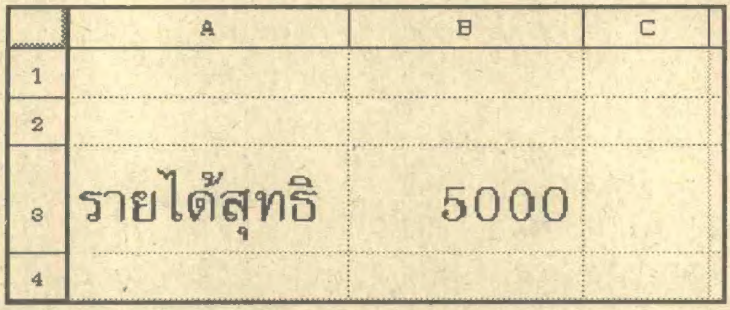
5. วางแผนการออกแบบชุดคำสั่งมาโคร

ก่อนการลงมือเขียนคำสั่งมาโครควรวางแผนการเขียนเสียก่อน เทคนิคการออกแบบที่ใช้ในการออกแบบก็มี ผังงาน (Flowchart), Pseudocode หรือ Hierarchy Chart ช่วยให้แผนงานประสพความสำเร็จ ในที่นี้จะกล่าวถึงการเขียนแผนงานเท่านั้น การเขียนผังงานทำให้เรามองเห็นภาพเส้นทางเดินของโปรแกรมหรือมาโครอย่างกว้าง ๆ และยังทำให้เกิดความรวดเร็วในการตรวจสอบจุดผิดพลาดที่เกิดขึ้นว่าทำไมผลลัพธ์ที่ได้ไม่ตรงกับความต้องการ ผังงานที่ดีนั้นจะต้องบันทึกหรือระบุเป้าหมายของชุดคำสั่งมาโครเอาไว้ด้วยเพื่อยึดเป็นหลัก เพราะบางครั้งมาโครที่สร้างขึ้นมีขนาดใหญ่ผู้เขียนอาจหลงเป้าได้

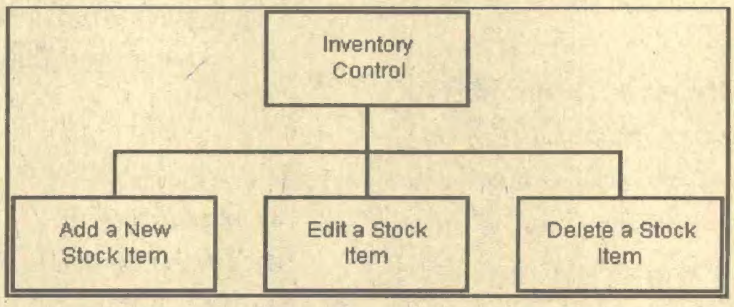
ชุดคำสั่งมาโครหรือโปรแกรมที่ดีนั้นควรประกอบด้วย 2 ส่วนหลักคือ มาโครควบคุม (Control Macro) และมาโครย่อย (Subroutine Macros) มาโครควบคุมทำหน้าที่ในการจ่ายงานย่อย ๆ ไปยังมาโครย่อย ในขณะที่มาโครย่อยทำหน้าที่เฉพาะกิจให้สำเร็จลุล่วงตามหน้าที่ที่ระบุไว้ เช่น การพิมพ์หัวกระดาษ (Heading) การหาผลรวม (Total) การ Acess Table เป็นต้น หลักการสร้างมาโครย่อยคือทำให้สั้นที่สุดเท่าที่จะทำได้ เพราะมาโครยิ่งสั้นยิ่งทำให้การเขียนคำสั่งง่ายขึ้น การตรวจสอบความผิดพลาด (Debugging) ง่ายขึ้น อีกทั้งการปรับปรุงแก้ไขก็ง่ายขึ้นเช่นกัน

ตัวอย่างที่แสดงในรูปที่ 3 และ 4 เป็นมาโครควบคุม (Control Macro) และมาโครย่อย (Subroutine Macros) ตามลำดับ รูปทั้ง 2 เป็นเพียงผังงานสาธิตเท่านั้น

ถึงแม้ว่าจะมีผังงานแล้วก็ตาม



รูปที่ 2 การระบุข้อความกำกับสูตร



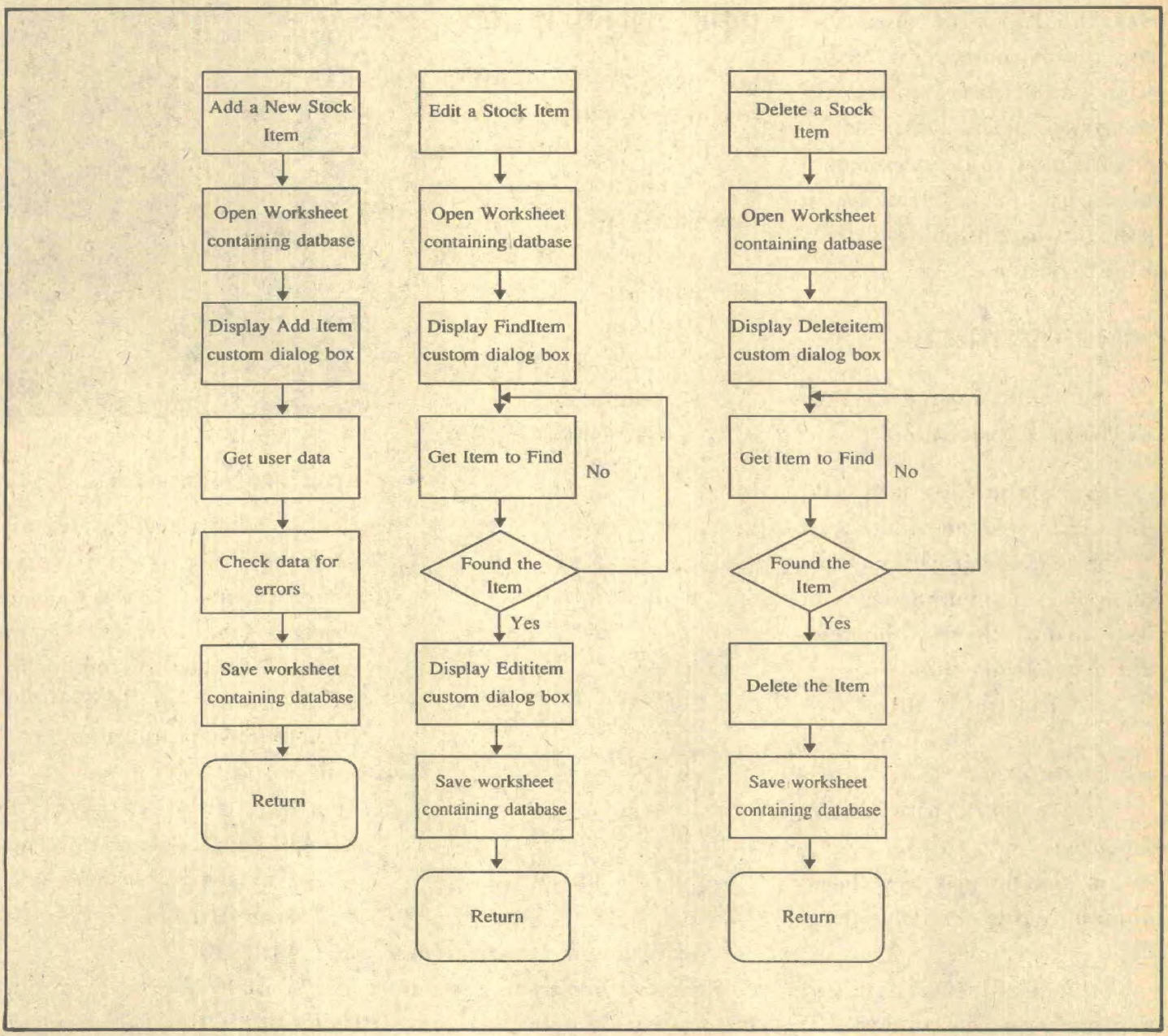
รูปที่ 3 มาโครควบคุม (Control Macro)

หน้าที่อีกประการหนึ่งของผู้เขียนมาโครก็คือ ผู้สร้างมาโครจะต้องเตรียมตั้งชื่อมาโครและการตั้ง Shortcut Key ไว้ด้วย การตั้งชื่อมาโครก็ควรให้มีชื่อสอดคล้องกับงานที่ทำด้วยจะเป็นการดี ส่วน Shortcut Key เหมาะสำหรับมาโครที่ต้องใช้งานบ่อย ๆ

การจัดฟอร์แมตให้กับมาโคร

มาโครที่แสดงในรูปที่ 1 เป็นมาโครที่ได้จัดหน้าตาให้ดูสะอาดแล้ว เคล็ดลับก็คือมาโครชีตใช้งานเพียงแค่ 3 คอลัมน์โดยที่

- คอลัมน์แรกแสดงชื่อ Shortcut Key เก็บรายละเอียดของ Shortcut Key เอาไว้
- คอลัมน์ที่สองแสดงชื่อของมาโครและสูตรคำนวณ
- คอลัมน์ที่สุดท้ายแสดงหมายเหตุ เป็นคอลัมน์ที่ใช้สำหรับบันทึกสิ่งต่าง ๆ ที่จำเป็น ผู้เขียนขอแนะนำให้



รูปที่ 4 มาโครย่อย (Subroutine Macro)

ทุกท่านพยายามใช้คอลัมน์นี้ให้เป็นประโยชน์ให้มากที่สุด

จะสังเกตเห็นว่าไม่มี Gridlines แสดงอยู่ที่เป็นเช่นนี้เพราะว่า Gridlines มีสถานะภาพเป็น Off มีการตีกรอบล้อมรอบคำสั่งมาโครเพียงเท่านี้มาโครชีตก็จะน่าดูยิ่งขึ้น

ถึงแม้ว่าการจัดหน้าตาให้มาโครชีตจะไม่ทำให้เวลาในการรันเร็วขึ้นก็ตาม งานที่มีขนาดใหญ่ ๆ การจัดหน้าตามาโครชีตดังตัวอย่างข้างต้นจะทำให้มาโครชีตดูดี และเมื่อถึงวาระรื้อระบบเวลาที่ใช้ในการไล่ที่มาของแต่ละส่วนคงจะไม่มากนัก เมื่อเปรียบเทียบกับการเขียนมาโครแบบไม่เป็นระเบียบที่ไว้ที่โน่นบ้างที่นี่บ้าง

หมวดคำสั่งพื้นฐาน

หมวดคำสั่งพื้นฐานสามารถแบ่งออกได้เป็น 6 หมวดดังนี้คือ

1. **คำสั่งในการกำหนดค่า** เป็นคำสั่งที่ใช้ในการกำหนดค่าให้กับตัวแปร

2. **คำสั่งควบคุมในโปรแกรม (Branching)** เป็นกลุ่มคำสั่งที่จะดำเนินการตามเงื่อนไขที่กำหนดให้แบ่งออกเป็น 3 กลุ่มย่อยด้วยกันคือ

2.1 คำสั่งเงื่อนไข (IF)

2.2 คำสั่งกระโดดไปทำยังส่วนต่าง ๆ (Goto)

2.3 คำสั่งวนรอบหรือทำซ้ำ (Looping)

3. **คำสั่งรับข้อมูลจากผู้ใช้ (Input)** เป็นกลุ่มคำสั่งที่ใช้ในการรับข้อมูลจากผู้ใช้

4. **คำสั่งพิเศษและการนำเอาฟังก์ชันมาใช้งาน** เป็นการนำเอาฟังก์ชันและคำสั่งพิเศษมาใช้งาน

5. **คำสั่งจัดการเกี่ยวกับแฟ้มข้อมูล** เป็นกลุ่มคำสั่งที่ใช้ในการอ่านข้อมูลจากไฟล์และเขียนข้อมูลลงไฟล์

ในบทความนี้จะเสนอรายละเอียดของแต่ละคำสั่งพร้อมทั้งตัวอย่างการใช้งานเพื่อให้ผู้อ่านสามารถประยุกต์ใช้งานได้

กลุ่มคำสั่งเงื่อนไข (IF)

กลุ่มคำสั่งนี้มีความสำคัญมาก เพราะชีวิตมนุษย์เต็มไปด้วยเงื่อนไข ถ้าคอมพิวเตอร์จะเป็นผู้ช่วยที่ดีของมนุษย์ก็จะต้องสามารถทำงานภายใต้เงื่อนไขต่าง ๆ ให้ได้

กลุ่มคำสั่งของ IF มีด้วยกัน 3 รูปแบบดังนี้คือ

- **รูปแบบที่ 1**

```
=IF(เงื่อนไข จริง เท็จ)
```

- **รูปแบบที่ 2**

```
=IF(เงื่อนไข)
  คำสั่ง...
  คำสั่ง...
  คำสั่ง...
=END.IF()
```

- **รูปแบบที่ 3**

```
=IF(เงื่อนไข)
  คำสั่ง...
  คำสั่ง...
=ELSE.IF(เงื่อนไข)
  คำสั่ง...
  คำสั่ง...
=ELSE()
  คำสั่ง...
  คำสั่ง...
=END.IF()
```

โดยที่รูปแบบที่ 1 สามารถใช้กับเวอร์กชีตและมาโครชีต ส่วน 2 รูปแบบหลังสามารถใช้ได้เฉพาะในมาโครชีตเท่านั้น

การประยุกต์ใช้งานคำสั่ง IF

ตัวอย่างการประยุกต์การใช้งานนี้จะเป็นการใช้คำสั่ง IF ในการตรวจสอบสถานภาพของ Row & Column Headings ถ้าอยู่ในสถานภาพ On ให้เปลี่ยนเป็น Off และในทางกลับกัน ถ้าอยู่ในสถานภาพ Off ให้เปลี่ยนเป็น On มีขั้นตอนในการดำเนินการหลัก ๆ ดังนี้

1. สร้างมาโครซึ่งทำหน้าที่ On หรือ Off Row & Column Headings

2. ตรวจสอบสถานภาพของ Row & Column Headings โดยใช้คำสั่ง GET.WINDOW

3. นำผลที่ได้จากข้อ 2 มาเป็นเงื่อนไขในคำสั่ง IF เพื่อสั่งให้ Headings เป็น On หรือ Off

1. สร้างมาโครชื่อ OnOffRowColHeadings

ก่อนประยุกต์ใช้คำสั่ง IF เราจะต้องสร้างมาโครขึ้นมาเสียก่อน ให้ชื่อมาโครว่า OnOffRowColHeadings ในครั้งนี้เราจะกำหนดตำแหน่งที่จะวางคำสั่งมาโครให้อีกด้วยโดยใช้คำสั่ง Macro Set Recorder หรือในภาคภาษาไทยเรียกว่า กำหนดตัวบันทึก ขั้นตอนมีดังนี้

การกำหนดตัวบันทึก

1. ไปที่เมนู File เลือก New เลือก Macro Sheet

2. ขณะนี้ท่านอยู่ในมาโครชีตชื่ออะไรไม่ต้องสนใจ ให้คลิ๊กเม้าส์ที่เซล B10

3. ไปที่เมนู Macro เลือก Set Recorder หมายความว่ามาโครที่เราจะบันทึกต่อไปนี้จะเริ่มต้นบันทึกที่เซล B10 เป็นต้นไป

การบันทึกมาโคร OnOffRow ColHeadings

1. จากเมนู Macro เลือก Start Recorder เพื่อเริ่มทำการบันทึกมาโคร

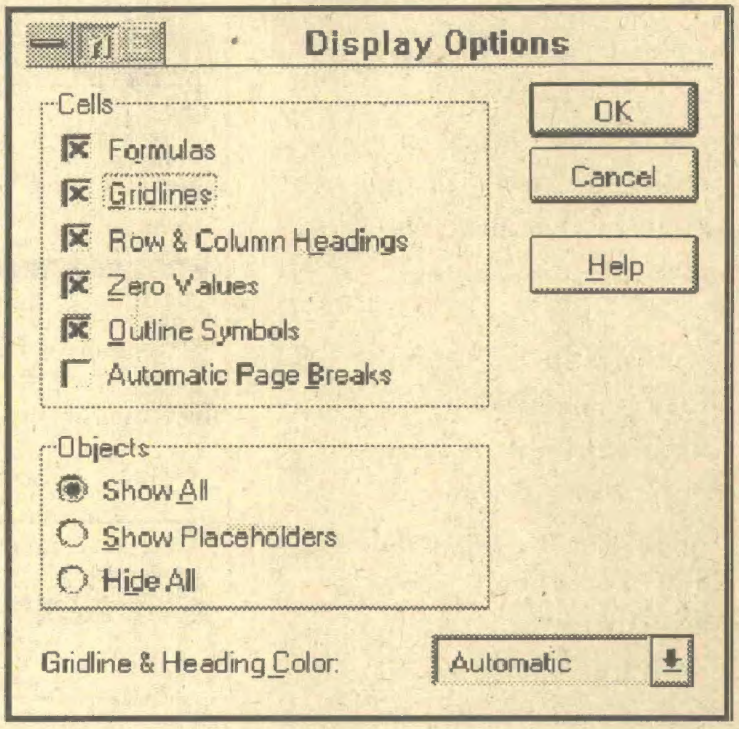
2. ไปที่เมนู Options เลือก Display หน้าจอจะปรากฏกรอบโต้ตอบดังแสดงในรูปที่ 5

3. คลิ๊กเม้าส์ที่หน้าหัวข้อ Row & Column Headings ถึงขณะนี้ถ้าทุกสิ่งทุกอย่างถูกต้องที่หน้าหัวข้อดังกล่าวจะไม่มีเครื่องหมาย X ปรากฏให้เห็น

4. คลิ๊กเม้าส์ที่ปุ่ม OK

5. ไปที่เมนู Macro เลือก Stop Recorder เพื่อยุติการบันทึกมาโคร

6. จะสังเกตเห็นว่าที่มาโครชีตไม่มี Row & Column Headings แสดงอยู่เลย



รูปที่ 5 กรอบโต้ตอบ Display Options

```
=DISPLAY(TRUE,TRUE,FALSE,TRUE,0,,TRUE,FALSE,1)
=RETURN()
```

รูปที่ 6 ฟังก์ชัน Display ที่ปรากฏอยู่ในมาโครชีต

ศึกษารายละเอียดของคำสั่ง

1. คลิ๊กเม้าส์ไปที่ตำแหน่งที่เก็บสูตรในที่นี้ก็คือ B10 จะได้สูตรดังแสดงในรูปที่ 6

2. จะเห็นว่าอาร์กิวเมนต์ของฟังก์ชัน DISPLAY มีมากมายเกินกว่าจะเดาได้ (แต่ถ้าจะเดากันจริง ๆ ก็พอจะเดากันได้)

3. เราสามารถตรวจสอบความหมายของอาร์กิวเมนต์ในฟังก์ชันได้ 2 วิธีคือ ศึกษาจากหนังสือ Function Reference เป็นวิธีแรก ส่วนวิธีที่สองคือตรวจสอบรายละเอียดจากคำสั่ง Past Function

การตรวจสอบความหมายของอาร์กิวเมนต์ในฟังก์ชัน Display ด้วย Past Function

1. คลิ๊กเม้าส์ที่ B10

2. ไปที่เมนู Formula เลือก Paste Function หน้าจอจะปรากฏกรอบโต้ตอบ Paste Function ดังแสดงในรูปที่ 7

3. ที่กรอบโต้ตอบ Paste Function ในกรอบ Function Category คลิ๊ก Commands ส่วนที่กรอบ Paste Function เลือก DISPLAY จากนั้นคลิ๊กปุ่ม OK ที่หน้าจอจะแสดงกรอบโต้ตอบย่อย Paste Function รายละเอียดของอาร์กิวเมนต์ดังแสดงในรูปที่ 8

จะเห็นว่าฟังก์ชัน DISPLAY แบ่งออกเป็น 2 รูปแบบคือ รูปแบบแรก (อาร์กิวเมนต์ที่แสดงอยู่ที่บรรทัดบนสุด) ใช้กับเวอร์กชีต ส่วนรูปแบบที่สอง (อาร์กิวเมนต์ที่แสดงอยู่ที่บรรทัดที่สอง) ใช้กับ Special Info Window ในที่นี้เราใช้รูปแบบแรก

4. อาร์กิวเมนต์ที่เราสนใจคือ headings อยู่ในลำดับที่ 3 ของบรรทัดแรก ซึ่งตรงกับลำดับที่แสดงในกรอบโต้ตอบ Display Options ดังแสดงในรูปที่ 5 และเมื่อสังเกตที่คำสั่งมาโครในรูป 6 จะเห็นว่าอาร์กิวเมนต์ในตำแหน่งที่ 3 มีค่าเป็น FALSE ในขณะที่อาร์กิวเมนต์ Gridlines มีค่าเป็น TRUE ถึงตอนนี้ที่กรอบโต้ตอบในรูปที่ 8 ให้คลิ๊กเม้าส์ที่ปุ่ม OK ที่ Formula Bar จะปรากฏฟังก์ชัน DISPLAY ให้คลิ๊กเม้าส์ที่เครื่องหมาย x หรือกดปุ่ม Esc

5. จากข้อ 4 เราสามารถตีความฟังก์ชัน DISPLAY ได้ดังนี้

```
= DISPLAY (,,TRUE)
```

หมายถึงให้แสดง Row & Column Headings

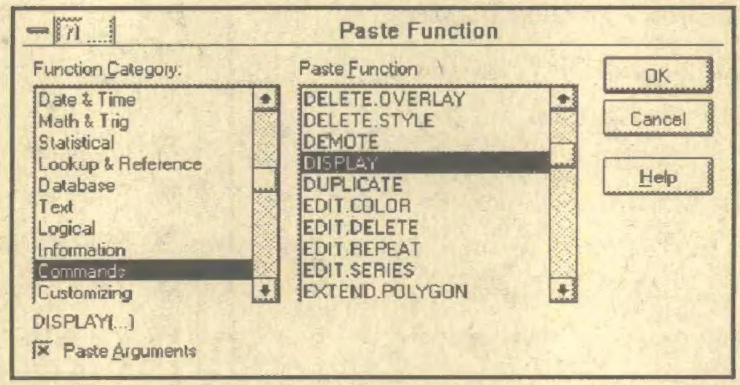
ส่วนคำสั่ง

```
= DISPLAY (,,FALSE)
```

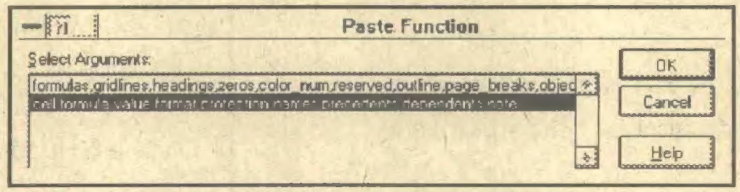
หมายถึงไม่มีการแสดง Row & Column Headings

6. ดังนั้นให้แก้ไขคำสั่งมาโครที่เซล B10 เสียใหม่ให้เป็นดังนี้

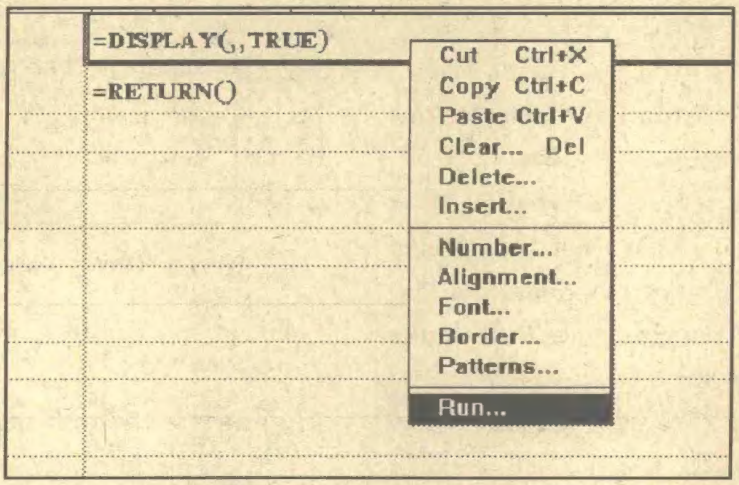
```
= DISPLAY (,,TURE)
```



รูปที่ 7 กรอบโต้ตอบ Paste Function



รูปที่ 8 แสดงอาร์กิวเมนต์ของฟังก์ชัน DISPLAY



รูปที่ 9 การรันมาโครผ่าน Shortct Key

รันมาโคร

1. คลิ๊กที่เซล B10 แล้วกดปุ่มข้างขวาของเม้าส์ จะมีเมนูแสดงขึ้นมาดังรูปที่ 9 จากนั้นให้คลิ๊กที่รายการ Run

2. ที่หน้าจอจะปรากฏกรอบโต้ตอบ Run Macro คล้ายคลึงกับที่แสดงในรูปที่ 10 จากนั้นให้คลิ๊กปุ่ม OK

3. จะสังเกตเห็นว่า Row &

Column Headings ของมาโครชีตปรากฏให้เห็นแล้วดังเดิม

2. ตรวจสอบสถานภาพของ Row & Column Headings โดยใช้คำสั่ง GET.WINDOW

ด้วยคำสั่ง GET.WINDOW จะทำให้เราสามารถตรวจสอบสถานภาพของ Row & Column Headings ได้ จากนั้นเราจะเก็บสถานภาพที่ได้ให้อยู่ในรูปตัวแปรเพื่อใช้ในคำสั่ง IF ในลำดับต่อไป

รูปแบบของคำสั่งคือ

GET.WINDOW (หมายเลข, ชื่อ window)

เมื่อหมายเลขคือหมายเลขของข้อมูลในวินโดว์ที่เราต้องการทราบ เช่น

9 คือสถานภาพของ Gridlines ถ้าปรากฏให้เห็นจะให้ค่า TRUE และในทางตรงกันข้ามจะให้ค่า FALSE

10 คือสถานภาพของ Row & Column Headings ถ้าปรากฏให้เห็นจะให้ค่า TRUE และในทางตรงกันข้ามจะให้ค่า FALSE

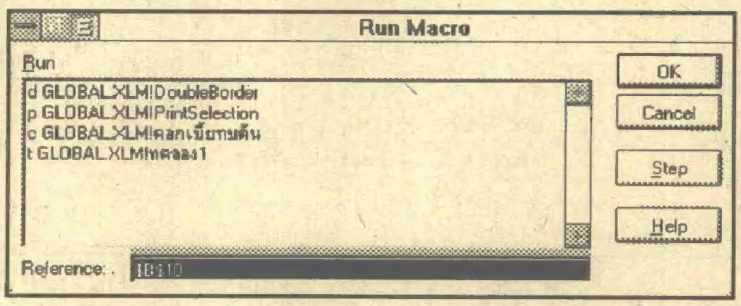
ชื่อวินโดว์คือชื่อของวินโดว์ที่เราต้องการทราบข้อมูล ถ้าละไว้หมายถึงต้องการทราบข้อมูลของวินโดว์ปัจจุบัน

แก้ไขมาโคร

เมื่อทราบความหมายของคำสั่ง GET.WINDOW แล้วเราจะทำการแก้ไขมาโครเสียใหม่ให้เป็นดังที่แสดงในรูปที่ 11

รันมาโครทดสอบอีกครั้ง

รันมาโครอีกครั้งแล้วกดแป้น Ctrl+' เพื่อดูผลลัพธ์ดังแสดงในรูปที่ 12



รูปที่ 10 กรอบโต้ตอบ Run Macro

	A	B
8		
9		
10		=GET.WINDOW(10)
11		=DISPLAY(,,TRUE)
12		=RETURN()
13		

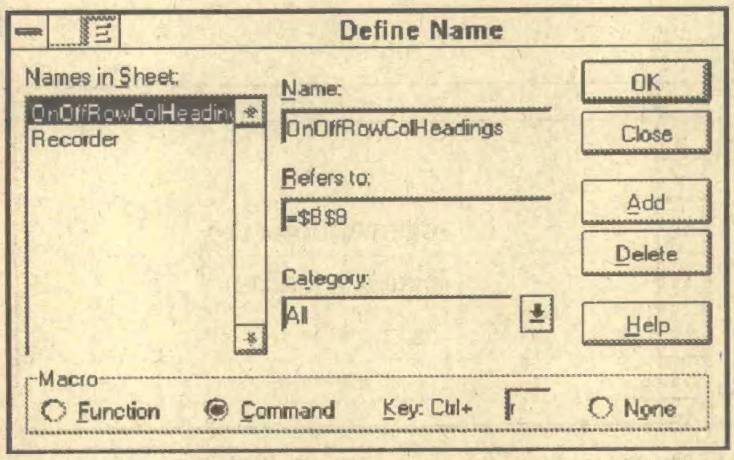
รูปที่ 11 แสดงมาโครหลังเพิ่มคำสั่ง GET.WINDOW

	A	B
8		
9		
10		TRUE
11		TRUE
12		TRUE
13		
14		

รูปที่ 12 ผลที่ได้จากการรันแล้วกด Ctrl+'

	A	B
8		
9		
10		=GET.WINDOW(10)
11		=IF(B10=TRUE,DISPLAY(,,FALSE),DISPLAY(,,TRUE)
12		=RETURN()
19		

รูปที่ 13 มาโครหลังจากเพิ่มคำสั่ง IF ลงไปแล้ว



รูปที่ 14 การตั้งชื่อมาโคร

3. นำผลที่ได้จากข้อ 2 มาเป็นเงื่อนไขในคำสั่ง IF เพื่อสั่งให้ Headings เป็น On หรือ Off

คราวนี้ก็ถึงตอนใช้คำสั่ง IF แล้วครับ เราจะใช้คำสั่ง IF รูปแบบแรก สิ่งที่เราต้องการนั้นสามารถเขียนเป็นภาษาที่เราเข้าใจกันได้ (ก่อนที่จะเขียนเป็นคำสั่งมาโคร) ดังนี้

IF(Row & Column Headings ปรากฏให้เห็น ให้จัดการ Off; ให้จัดการ On)

แก้ไขมาโครเสียใหม่ให้เป็นดังรูปที่ 13

ตั้งชื่อและตั้ง Shortcut Key ให้กับมาโคร

1. คลิ๊กที่ B8 พิมพ์ OnOff-RowColHeadings

2. จัดตัวอักษรให้เป็นตัวเข้ม (Bold)

3. ขณะนี้ B8 ยังถูกเลือกอยู่ ลำดับต่อไปเราจะทำการตั้งชื่อ ไปที่เมนู Formula แล้วเลือกคำสั่ง Define Name

4. หน้าจอจะปรากฏกรอบโต้ตอบ Define Name คล้ายคลึงกับในรูปที่ 14 ตั้งชื่อ Shotcut Key ว่า Cut + r ส่วนชื่อมาโครคือ OnOffRowCol-Headings

5. จัดการเซฟไฟล์นี้ให้ชื่อว่า EX21 ไปที่เมนู File เลือก Save แล้วจัดการใส่ชื่อไฟล์ดังกล่าวให้เรียบร้อย

พบกันใหม่ฉบับหน้า

เมนู...เล่มล่า...

- การติดตั้งระบบ Config.sys และ Config.db 7
- จากดีเบสทรีไปเป็นดีเบสทรีพลัส 15
- การใช้ CATALOG 31
- การเขียนโปรแกรมด้วยดีเบสทรีพลัส 35
- เรื่องของตัวแปรและอะเรย์ 60
- ดีเบสกับเอ็มเอสดอส 70
- ความสัมพันธ์ระหว่างไฟล์ 77
- เวลาเป็นเงินเป็นทอง 87
- e^x log N และค่าทางตรีโกณมิติ 94
- ดีเบสกับการสร้างกราฟ 104
- ใช้ดีเบสทรีจัดเรียงคำภาษาไทย Thai Sort 122
- ดีเบสทรีพลัสกับภาษาแอสเซมบลี 131
- โปรแกรมจัดโครงสร้างโปรแกรม (Automatic Indent Lister: AIL version 1.01) 151

ฯลฯ

อิ่มเพียง 110 บาท

จาก

ไมโครคอมพิวเตอร์